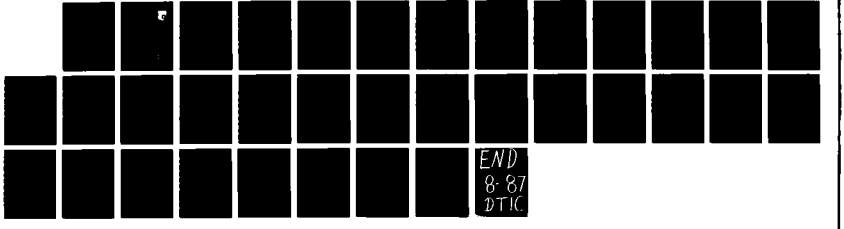
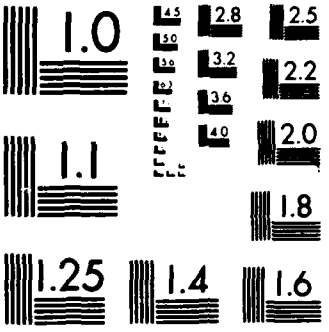
AD-A182 650 INTEGRATED INFORMATION SUPPORT SYSTEM (IISS) VOLUME 8 1/1
USER INTERFACE SUBS..(U) GENERAL ELECTRIC CO
SCHENECTADY NY PRODUCTION RESOURCES CONSU..
UNCLASSIFIED C MORENC ET AL. 01 NOV 85 UTP-620144501 F/G 12/5 NL





AFWAL-TR-86-4006
Volume VIII
Part 23

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 23 - Report Writer Unit Test Plan

General Electric Company
Production Resources Consulting
One River Road
Schenectady, New York 12345

Final Report for Period 22 September 1980 - 31 July 1985

November 1985

MATERIALS LABORATORY
AIR FORCE WRIGHT AERONAUTICAL LABORATORIES
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AFB, OH 45433-6533

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever; and the fact that the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data, is not to be regarded by implication or otherwise as in any manner licensing the holder or any other person or corporation, or conveying any rights or permission to manufacture, use, or sell any patented invention that may in any way be related thereto.

This report has been reviewed by the Office of Public Affairs (ASD/PA) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nations.

This technical report has been reviewed and is approved for publication.

DAVID L. JUDSON, PROJECT MANAGER
AFWAL/MLTC
WRIGHT PATTERSON AFB OH 45433

5 Aug 1986
DATE

FOR THE COMMANDER:

GERALD C. SHUMAKER, BRANCH CHIEF
AFWAL/MLTC
WRIGHT PATTERSON AFB OH 45433

7 Aug 86
DATE

"If your address has changed, if you wish to be removed from our mailing list, or if the addressee is no longer employed by your organization please notify AFWAL/MLTC, W-PAFB, OH 45433 to help us maintain a current mailing list."

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

A182 650

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release: distribution is unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S)	5. MONITORING ORGANIZATION REPORT NUMBER(S) AFWAL-TR-86-4006 Vol VIII, Part 23
6a. NAME OF PERFORMING ORGANIZATION General Electric Company Production Resources Consulting 6b. OFFICE SYMBOL (If applicable)	7a. NAME OF MONITORING ORGANIZATION AFWAL/MLTC
6c. ADDRESS (City, State and ZIP Code) 1 River Road Schenectady, NY 12345	7b. ADDRESS (City, State and ZIP Code) WPAFB, OH 45433-6533
8a. NAME OF FUNDING/SPONSORING ORGANIZATION Materials Laboratory Air Force Systems Command, USAF 8b. OFFICE SYMBOL (If applicable) AFWAL/MLTC	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER F33615-80-C-5155
8c. ADDRESS (City, State and ZIP Code) Wright-Patterson AFB, Ohio 45433	10 SOURCE OF FUNDING NOS PROGRAM ELEMENT NO. 78011F; PROJECT NO. 7500; TASK NO. 62; WORK UNIT NO. 01
11. TITLE (Include Security Classification) (See Reverse)	
12. PERSONAL AUTHOR(S) Glandorf, Frank and Morenc, Carol and Robie, Penny	
13a. TYPE OF REPORT Final Technical Report 13b. TIME COVERED 22 Sept 1980 - 31 July 1985	14. DATE OF REPORT (Yr., Mo., Day) 1985 November 15. PAGE COUNT 38
16. SUPPLEMENTARY NOTATION ICAM Project Priority 6201	The computer software contained herein are theoretical and/or references that in no way reflect Air Force-owned or -developed computer software.

17 COSATI CODES		
FIELD	GROUP	SUB GR.
1308	0905	

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This unit test plan establishes the methodology and procedures used to adequately test the capabilities of the computer program identified as the Report Writer (RW). The RW Application Generator is used to translate report definitions into programs that access data bases via the Common Data Model and report the extracted data in a formatted way usually with interspersed identifying text and possible statistical summaries. The destination of the reports is some hardcopy medium such as line printer output.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT. ☐ DTIC USERS ☐	21 ABSTRACT SECURITY CLASSIFICATION Unclassified
22a NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b TELEPHONE NUMBER (Include Area Code) 513-255-6976 22c OFFICE SYMBOL AFWAL/MLTC

DD FORM 1473, 83 APR EDITION OF 1 JAN 73 IS OBSOLETE.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE

11. Title

Integrated Information Support System (IISS)
Vol VIII - User Interface Subsystem
Part 23 - Report Writer Unit Test Plan

A S D 86 0038
9 Jan 1986

PREFACE

This unit test plan covers the work performed under Air Force Contract F33615-80-C-5155 (ICAM Project 6201). This contract is sponsored by the Materials Laboratory, Air Force Systems Command, Wright-Patterson Air Force Base, Ohio. It was administered under the technical direction of Mr. Gerald C. Shumaker, ICAM Program Manager, Manufacturing Technology Division, through Project Manager, Mr. David Judson. The Prime Contractor was Production Resources Consulting of the General Electric Company, Schenectady, New York, under the direction of Mr. Alan Rubenstein. The General Electric Project Manager was Mr. Myron Hurlbut of Industrial Automation Systems Department, Albany, New York.

Certain work aimed at improving Test Bed Technology has been performed by other contracts with Project 6201 performing integrating functions. This work consisted of enhancements to Test Bed software and establishment and operation of Test Bed hardware and communications for developers and other users. Documentation relating to the Test Bed from all of these contractors and projects have been integrated under Project 6201 for publication and treatment as an integrated set of documents. The particular contributors to each document are noted on the Report Documentation Page (DD1473). A listing and description of the entire project documentation system and how they are related is contained in document FTR620100001, Project Overview.

The subcontractors and their contributing activities were as follows:

TASK 4.2

Subcontractors	Role
Boeing Military Aircraft Company (BMAC)	Reviewer.
D. Appleton Company (DACOM)	Responsible for IDEF support, state-of-the-art literature search.
General Dynamics/ Ft. Worth	Responsible for factory view function and information models.

Subcontractors	Role
Illinois Institute of Technology	Responsible for factory view function research (IITRI) and information models of small and medium-size business.
North American Rockwell	Reviewer.
Northrop Corporation	Responsible for factory view function and information models.
Pritsker and Associates	Responsible for IDEF2 support.
SofTech	Responsible for IDEF0 support.

TASKS 4.3 - 4.9 (TEST BED)

Subcontractors	Role
Boeing Military Aircraft Company (BMAC)	Responsible for consultation on applications of the technology and on IBM computer technology.
Computer Technology Associates (CTA)	Assisted in the areas of communications systems, system design and integration methodology, and design of the Network Transaction Manager.
Control Data Corporation (CDC)	Responsible for the Common Data Model (CDM) implementation and part of the CDM design (shared with DACOM).
D. Appleton Company (DACOM)	Responsible for the overall CDM Subsystem design integration and test plan, as well as part of the design of the CDM (shared with CDC). DACOM also developed the Integration Methodology and did the schema mappings for the Application Subsystems.

Subcontractors	Role
Digital Equipment Corporation (DEC)	Consulting and support of the performance testing and on DEC software and computer systems operation.
McDonnell Douglas Automation Company (McAuto)	Responsible for the support and enhancements to the Network Transaction Manager Subsystem during 1984/1985 period.
On-Line Software International (OSI)	Responsible for programming the Communications Subsystem on the IBM and for consulting on the IBM.
Rath and Strong Systems Products (RSSP) (In 1985 became McCormack & Dodge)	Responsible for assistance in the implementation and use of the MRP II package (PIOS) that they supplied.
SofTech, Inc.	Responsible for the design and implementation of the Network Transaction Manager (NTM) in 1981/1984 period.
Software Performance Engineering (SPE)	Responsible for directing the work on performance evaluation and analysis.
Structural Dynamics Research Corporation (SDRC)	Responsible for the User Interface and Virtual Terminal Interface Subsystems.

Other prime contractors under other projects who have contributed to Test Bed Technology, their contributing activities and responsible projects are as follows:

Contractors	ICAM Project	Contributing Activities
Boeing Military Aircraft Company (BMAC)	1701, 2201, 2202	Enhancements for IBM node use. Technology Transfer to Integrated Sheet Metal Center (ISMC).

Contractors	ICAM Project	Contributing Activities
Control Data Corporation (CDC)	1502, 1701	IISS enhancements to Common Data Model Processor (CDMP).
D. Appleton Company (DACOM)	1502	IISS enhancements to Integration Methodology.
General Electric	1502	Operation of the Test Bed and communications equipment.
Hughes Aircraft Company (HAC)	1701	Test Bed enhancements.
Structural Dynamics Research Corporation (SDRC)	1502, 1701, 1703	IISS enhancements to User Interface/Virtual Terminal Interface (UI/VTI).
Systran	1502	Test Bed enhancements. Operation of Test Bed.

TABLE OF CONTENTS

APPENDICES

FIGURES

SECTION 1

GENERAL

1.1 Purpose

This unit test plan establishes the methodology and procedures used to adequately test the capabilities of the computer program identified as the Report Writer known in this document as RW. The RW is a configuration item of the Integrated Information Support System (IISS) User Interface (UI).

1.2 Project References

[1] ICAM Documentation Standards, 15 September 1983, IDS150120000C.

[2] General Electric Co., System Design Specification, 7 February 1983.

[3] Structural Dynamics Research Corporation, Report Writer Development Specification, DS 620144501 , 1 November 1985.

[4] Structural Dynamics Research Corporation, Rapid Application Generator Unit Test Plan, UTP620144502 , 1 November 1985.

[5] Structural Dynamics Research Corporation, Text Editor Unit Test Plan, UTP620144600 , 1 November 1985.

[6] Structural Dynamics Research Corporation, Form Processor Unit Test Plan, UTP620144200 , 1 November 1985.

[7] Structural Dynamics Research Corporation, Application Interface Unit Test Plan, UTP620144700 , 1 November 1985.

[8] Structural Dynamics Research Corporation, Forms Language Compiler Unit Test Plan, UTP620144401 , 1 November 1985.

[9] Structural Dynamics Research Corporation, Forms Driven Form Editor Unit Test Plan, UTP620144402 , 1 November 1985.

[10] Structural Dynamics Research Corporation, User Interface Services Unit Test Plan, UTP620144100 , 1 November 1985.

[11] Structural Dynamics Research Corporation, Virtual Terminal Unit Test Plan, UTP620144300 , 1 November 1985.

1.3 Terms and Abbreviations

Application Generator: (AG), subset of the IISS User Interface that consists of software modules that generate IISS application code and associated form definitions based on a language input. The part of the AG that generates report programs is called the Report Writer. The part of the AG that generates interactive applications is called the Rapid Application Generator.

Application Interface: (AI), subset of the IISS User Interface that consists of the callable routines that are linked with applications that use the Form Processor or Virtual Terminal. The AI enables applications to be hosted on computers other than the host of the User Interface.

Application Process: (AP), a cohesive unit of software that can be initiated as a unit to perform some function or functions.

Attribute: field characteristic such as blinking, highlighted, black, etc. and various other combinations. Background attributes are defined for forms or windows only. Foreground attributes are defined for items. Attributes may be permanent, i.e., they remain the same unless changed by the application program, or they may be temporary, i.e., they remain in effect until the window is redisplayed.

Common Data Model: (CDM), IISS subsystem that describes common data application process formats, form definitions, etc. of the IISS and includes conceptual schema, external schemas, internal schemas, and schema transformation operators.

Computer Program Configuration Item: (CPCI), an aggregation of computer programs or any of their discrete portions, which satisfies an end-use function.

Conceptual Schema: (CS), the standard definition used for all data in the CDM. It is based on IDEF1 information modelling.

Device Drivers: (DD), software modules written to handle I/O for a specific kind of terminal. The modules map terminal specific commands and data to a neutral format. Device Drivers are part of the UI Virtual Terminal.

Display List: is similar to the open list, except that it contains only those forms that have been added to the screen and are currently displayed on the screen.

External Schema: (ES), an application's view of the CDM's conceptual schema.

Field: two-dimensional space on a terminal screen.

Form: structured view which may be imposed on windows or other forms. A form is composed of fields. These fields may be defined as forms, items, and windows.

Form Definition: (FD), forms definition language after compilation. It is read at runtime by the Form Processor.

Forms Definition Language: (FDL), the language in which electronic forms are defined.

Forms Driven Form Editor: (FDFE), subset of the FE which consists of a forms driven application used to create Form Definition files interactively.

Form Editor: (FE), subset of the IISS User Interface that is used to create definitions of forms. The FE consists of the Forms Driven Form Editor and the Forms Language Compiler.

Form Hierarchy: a graphic representation of the way in which forms, items and windows are related to their parent form.

Forms Language Compiler: (FLAN), subset of the FE that consists of a batch process that accepts a series of forms definition language statements and produces form definition files as output.

Form Processor: (FP), subset of the IISS User Interface that consists of a set of callable execution time routines available to an application program for form processing.

IISS Function Screen: the first screen that is displayed after logon. It allows the user to specify the function he wants to access and the device type and device name on which he is working.

Integrated Information Support System: (IISS), a test computing environment used to investigate, demonstrate and test the concepts of information management and information integration in the context of Aerospace Manufacturing. The IISS addresses the problems of integration of data resident on heterogeneous data bases supported by heterogeneous computers interconnected via a Local Area Network.

Item: non-decomposable area of a form in which hard-coded descriptive text may be placed and the only defined areas where user data may be input/output.

Message: descriptive text which may be returned in the standard message line on the terminal screen. They are used to warn of errors or provide other user information.

Message Line: a line on the terminal screen that is used to display messages.

Network Transaction Manager: (NTM), IISS subsystem that performs the coordination, communication and housekeeping functions required to integrate the Application Processes and System Services resident on the various hosts into a cohesive system.

Neutral Data Manipulation Language: (NDML), the command language by which the CDM is accessed for the purpose of extracting, deleting, adding, or modifying data.

Operating System: (OS), software supplied with a computer which allows it to supervise its own operations and manage access to hardware facilities such as memory and peripherals.

Page: instance of forms in windows that are created whenever a form is added to a window.

Paging and Scrolling: a method which allows a form to contain more data than can be displayed with provisions for viewing any portion of the data buffer.

Physical Device: a hardware terminal.

Presentation Schema: (PS), may be equivalent to a form. It is the view presented to the user of the application.

Qualified Name: the name of a form, item or window preceded by the hierarchy path so that it is uniquely identified.

Report Definition Language: an extension of the Forms Definition Language that includes retrieval and calculation of database information and is used to define reports.

Report Writer: (RW), part of the Application Generator that generates source code for report programs based on a language input.

Subform: a form that is used within another form.

Text Editor: (TE), subset of the IISS User Interface that consists of a file editor that is based on the text editing functions built into the Form Processor.

User Data: data which is either input by the user or output by the application programs to items.

User Interface: (UI), IISS subsystem that controls the user's terminal and interfaces with the rest of the system. The UI consists of two major subsystems: the User Interface Development System (UIDS) and the User Interface Management System (UIMS).

User Interface Development System: (UIDS), collection of IISS User Interface subsystems that are used by applications programmers as they develop IISS applications. The UIDS includes the Form Editor and the Application Generator.

User Interface Management System: (UIMS), the runtime UI. It consists of the Form Processor, Virtual Terminal, Application Interface, the User Interface Services and the Text Editor.

User Interface Services: (UIS), subset of the IISS User Interface that consists of a package of routines that aid users in controlling their environment. It includes message management, change password, and application definition services.

User Interface/Virtual Terminal Interface: (UI/VTI), another name for the User Interface.

Window: dynamic area of a terminal screen on which predefined forms may be placed at run time.

Window Manager: a facility which allows the following to be manipulated: size and location of windows, the device on which an application is running, the position of a form within a window. It is part of the Form Processor.

SECTION 2

DEVELOPMENT ACTIVITY

2.1 Statement of Pretest Activity

During system development, the computer program was tested progressively. Functionality was incrementally tested and as bugs were discovered by this testing, the software was corrected.

This testing was conducted by the individual program developer in a manual mode. Any errors were noted by the developer and corrections to the program were then made after a testing session.

2.2 Pretest Activity Results

Testing of the RW discovered a few minor bugs which were then corrected and retesting proved successful. Testing included exceptional conditions and error conditions for the language. The overall test results during development showed no major programming errors. Only minor bugs were discovered and corrected.

SECTION 3

SYSTEM DESCRIPTION

3.1 System Description

The Report Writer Application Generator is used to translate report definitions into programs that access data bases via the CDM and report the extracted data in a formatted way usually with interspersed identifying text and possibly statistical summaries. The destination of the reports is some hardcopy medium such as lineprinter output.

The Forms Definition Language in which the report definitions are expressed includes the Forms Definition Language and other statement types.

The COBOL program output by the RW is constrained to be compatible with statement forms expected by the CDM precompiler.

The syntax of the Application Definition Language accepted as input to FLAN is modelled after the Forms Definition Language and the Neutral Data Manipulation Language.

The interface block diagram for the Report Writer Application Generator is shown in Figure 3-1.

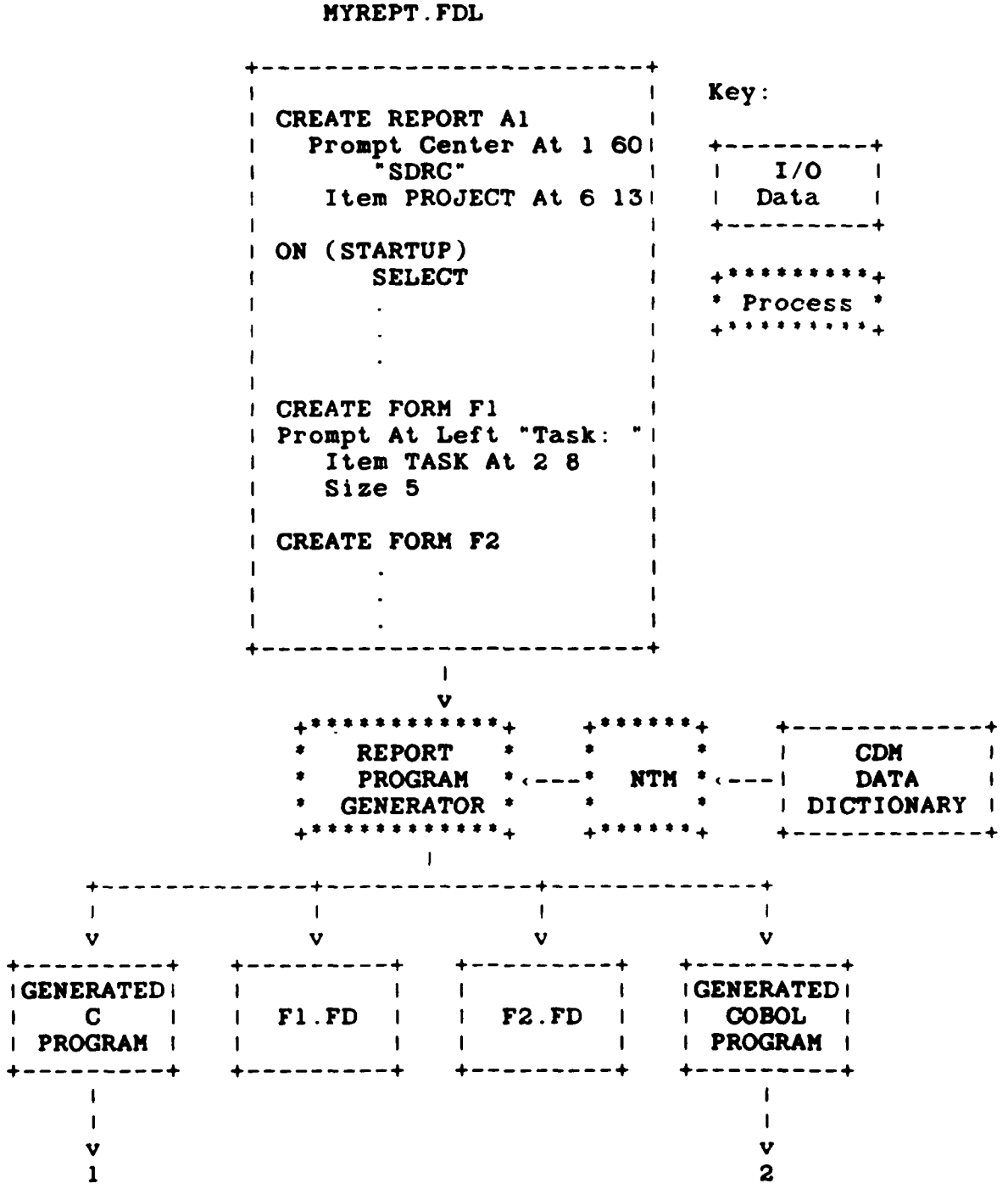

MYREPT.FDL
CREATE REPORT A1
Prompt Center At 1 60
"SDRC"
Item PROJECT At 6 13
ON (STARTUP)
SELECT
CREATE FORM F1
Prompt At Left "Task: "
Item TASK At 2 8
Size 5
CREATE FORM F2
Key:
I/O Data
Process
REPORT PROGRAM GENERATOR
NTM
CDM DATA DICTIONARY
GENERATED C PROGRAM
F1.FD
F2.FD
GENERATED COBOL PROGRAM
1
2

Fig. 3-1a

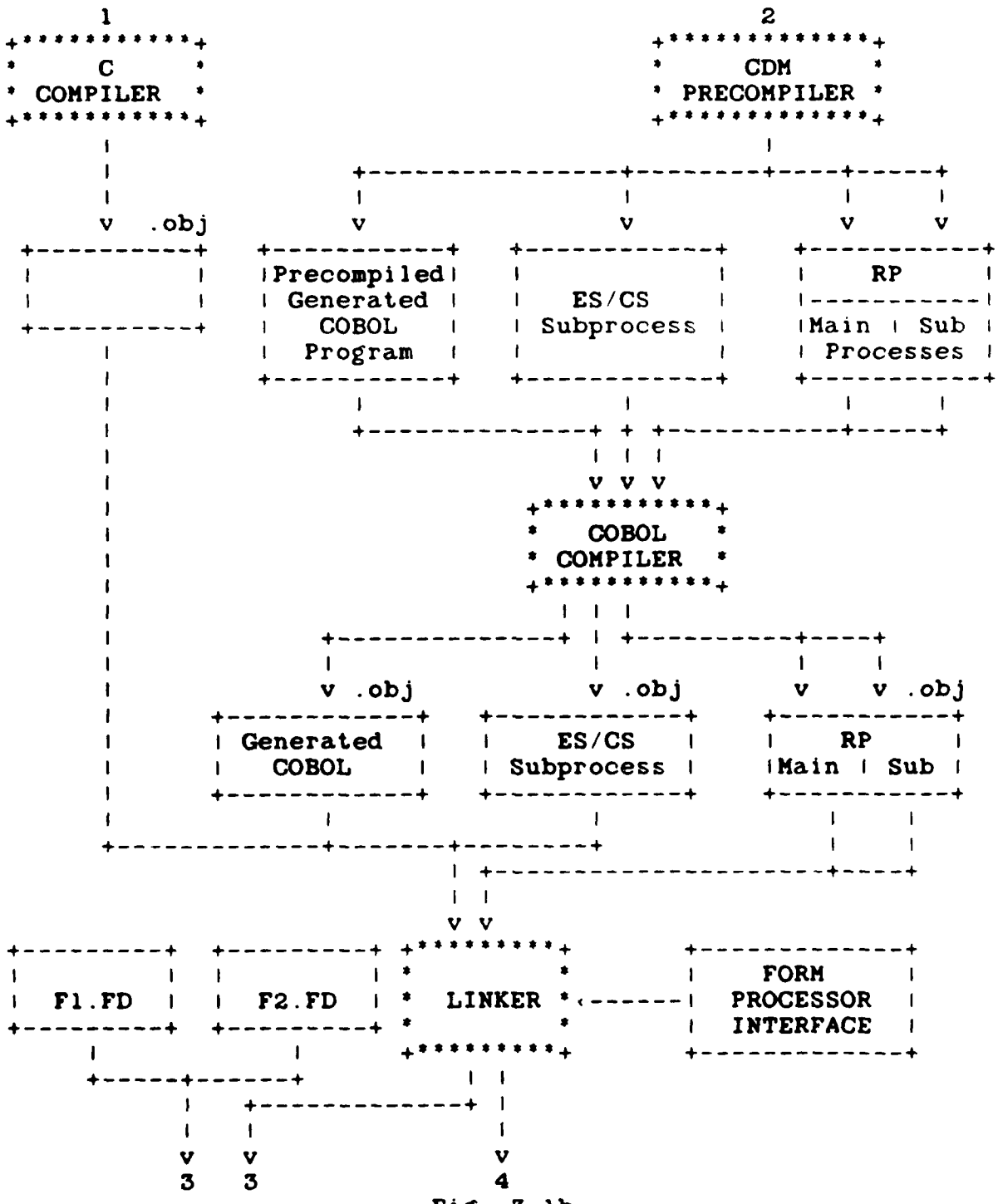

1
C
COMPILER
2
CDM
PRECOMPILER
.obj
Precompiled
Generated
COBOL
Program
ES/CS
Subprocess
RP
Main
Sub
Processes
COBOL
COMPILER
.obj
Generated
COBOL
.obj
ES/CS
Subprocess
.obj
RP
Main
Sub
F1.FD
F2.FD
LINKER
FORM
PROCESSOR
INTERFACE
3
3
4

Fig. 3-1b

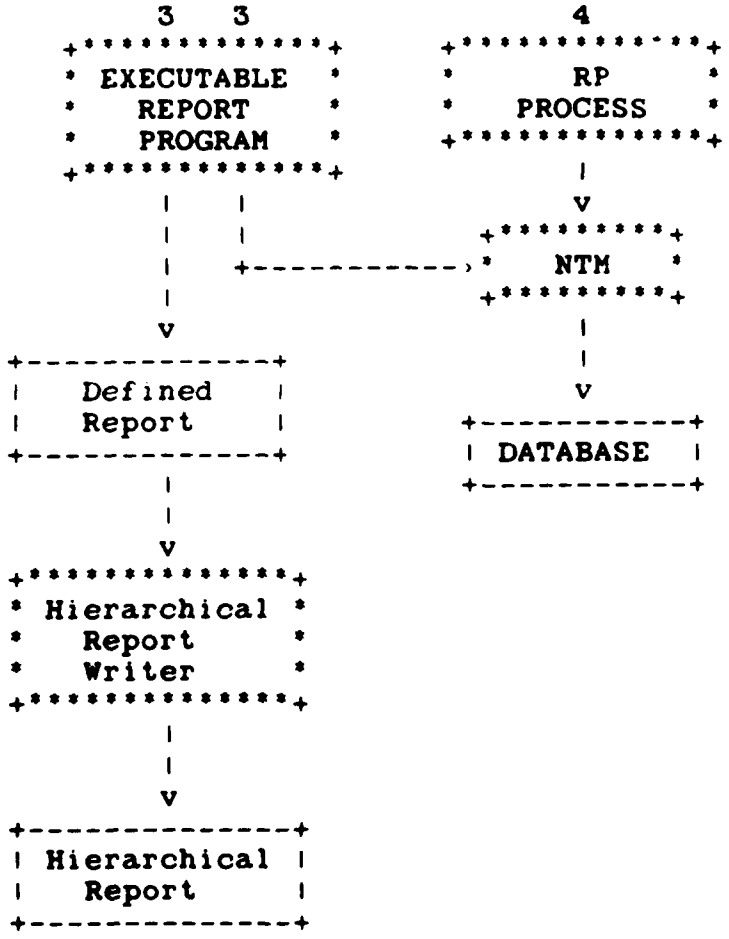


Figure 3-1c Report Writer Application Generator Interfaces

3.2 Testing Schedule

The execution of the Report Writer generator is dependent upon the CDM and NTM subsystems of IISS and testing of the RW must be done only after the CDM and NTM have themselves been successfully tested. Since COBOL code generated by the RW must be precompiled, the precompiler must also be tested prior to testing of the RW. Finally within the UI subsytem, the RW uses the Form Processor and FLAN and therefore must be tested only after their successful tests.

3.3 First Location Testing

These tests of the RW require the following:

Equipment: Air Force VAX.

Support Software: The Integrated Information Support System, the Oracle database management system, C compiler, COBOL compiler.

Personnel: One integrator familiar with the IISS.

Training: The Report Writer User manual for the current release.

Deliverables: The Report Writer Application Generator Subsystem of the IISS UI/VTI.

Test Materials: This test is interactive and can be manually performed as outlined in this test plan. No script file has been provided because scripting will not work with this essentially batch process.

Security considerations: None.

3.4 Subsequent Location Testing

The requirements as listed above need to be met. In future tests it will not be necessary to update the data base tables, NTM tables or use SDDEFINEAP, if the same report names are used. This Unit Test Plan useswas written for IISS Release 2.0 and may become obsolete for futer releases.

SECTION 4

SPECIFICATIONS AND EVALUATIONS

4.1 Test Specification

The following requirements are demonstrated by the outlined tests:

Functional Requirements	Test Activity A	B	C	D	E	F	G	H	I	J	K	L
Open ended array of forms.	*											
Nonduplication of items.		*										
Calculated fields.			*									
Initialization.				*								
Detect array overflow.					*							
Detect change of values.						*						
Page Breaks.							*					
Adding a form to a window.								*				
Adding an element to array.									*			
Setting a field to a value.										*		
Data base query.											*	

A - Fields qtype and stype.
B - Field dbname.
C - Fields pdate and ppage.
D - Startup condition.
E - Overflow condition.
F - Change condition.
G - Page action.
H - Present form action.
I - Present qualified name action.
J - Set action.
K - Select action.

The steps outlined in Section 5.3 and the file in Appendix A show the direct correspondence between the test and the functional requirements as listed in this section.

4.2 Testing Methods and Constraints

The tests as outlined in Section 5.3 must be followed. The required input is stated for each test. This testing tests the normal mode of operation of these functions and does not completely exercise all the error combinations that a user of

the RW might create by faulty entry of field information. These tests have been done, however, through the normal testing done by the developer of these functions. No additional constraints are placed on this unit test besides those listed in Section 3.3 of this unit test plan.

4.3 Test Progression

The progression of testing of the RW is fully outlined in Section 5.3 of this unit test plan. This progression should be followed exactly to insure the successful testing of this IISS configuration item.

4.4 Test Evaluation

The complete Report Writer Generator test consists of many stages each having its associated output. The first stage is the input and processing of the application definition by the generator. The outputs are generated C and COBOL files and the binary form files TSTRPT, QYTPE, and STYPE.

The second stage is the precompilation of the COBOL file. This should successfully produce four COBOL procedures. The names of these procedures and the names of the files containing them are constructed at generation time. The files names as well as the success or failure of the precompilation are reported to the test evaluator in another file named according to his choice. The procedure names must be found by looking within the procedures themselves.

The third stage is the compilation and linking of the code which has been created in stages one and two. The respective compilers and linker will report the success or failure of the steps comprising this stage of the test.

The fourth and final stage of the test is the execution of the generated application. The success of this stage will depend upon the successful operation of the NTM, the CDM, and the Form Processor. The resulting output for a successful test is in RPT1.DAT in the NTM environment directory. A comparison report may be found in Appendix B. The two reports may not have exactly the same data since the contents of the CDM may change between tests; however, the format of the report should be the same.

SECTION 5

TEST PROCEDURES

5.1 Test Description

A general description of this unit test is provided in Section 5.3.

5.2 Test Control

As outlined, this unit test is a manual test which may be done by anyone. The required input data are documented for each function being tested and the resulting successful output is also documented. The order of the testing is also completely documented. The test control information is completely described in Section 5.3.

5.3 Test Procedures

To run the unit test plan as outlined below, one must be logged on to an IISS account. The NTM must be up and running and the UI group logical names IISSFLIB, IISSULIB and IISSMLIB must be set properly. IISSFLIB points to the directory containing form definitions (.FD files). IISSULIB points to the NTM environment directory since the report writer writes the fd files out to the directory and in subsequent running of the report, these fd files are used. IISSMLIB points to the directory containing error messages (MSG files).

Below is an example of how the Report Writer may be invoked in the VAX/VMS environment for Release 2.0. This example requires the use of two terminals. In normal usage of the RW, if the NTM is already running, only one terminal is needed. The steps are numbered sequentially and those that are executed on the first terminal are indicated here with an "A", those on the second terminal are labeled with a "B". The following convention is used to document the example.

- Text in angle brackets is to be replaced with appropriate information by the user.
- Single upper case words enclosed in angle brackets represent terminal keys (e.g. ‹ENTER›).
- Text in upper case is to be entered as shown.
- Comments and instructions are in bold.

Stage 1 ---

1-A Logon on terminal A

2-A $ SET DEF ‹to directory containing your NTM environment›

3-A $ @IISS This brings up the NTM

4-B Logon on terminal B

5-B $ SET DEF ‹to directory containing your NTM environment›

6-B $ VT100 This starts up the VT100 device driver. If the User Interface system has been installed at your site with a different device driver, then this step is amended as appropriate.

7-B Fill in the items on the IISS logon screen as follows:

Username: ‹your username›
Password: ‹your password›
Role : ‹your role›
Press ‹ENTER›

8-B Fill in the function item on the function screen as follows:

Function: SDREPWRITR
Press ‹ENTER›

9-B Fill in the field on the report writer generator screen as follows, where [reptdir] is the directory where rpt1.fdl resides:

Application Name: [reptdir]rpt1.fdl
This compiles the application definition, creates the .FD files, and produces both the C code (rpt1C.C) and the COBOL code (rpt1X.PRC).

Press ‹QUIT› Returns you to the operating system.

Stage 2 ---

10-B $ @[NDML_dir]NDML This invokes the NDML precompiler on the test bed and produces as output four or more COBOL files named Xxxxx.TMP:

- database application
- one or more RP-SUB process files
- one or more CS-ES (conceptual_to_external_schema) subroutines
- RP-Main process

Inside these files are the Yxxxx process name files.

This step requires an input file containing precompiler directives. Please see the precompiler documentation for the format of this file.

Stage 3 ---

11-B $ @[NDML_dir]COMPANS ‹one of the Yxxxx files›
Repeat this step with each of the Xxxxx files that were generated. These steps compile each COBOL file.

12-B $ LIB/REP GENLIB.OLB *.OBJ
This step inserts the generated object modules in a library named genlib.olb in the user's directory.

13-B $ CC testapC This compiles the .C code.

14-B $ @[NDML_dir]LNKORP ‹RP-Main process name›
This step links the RP-Main to the RP-Sub to produce an executable.
NOTE: This RP process name is the Yxxxx name.

15-B $ @LNKAP testap This step links the compiled .C and .COB modules with the RAP library to produce an executable.

Stage 4 ---

16-B $ DELETE SEL*.DAT;* When executed, the generated application creates files named SELn.DAT. These remain after the report terminates. To avoid confusion it is recommended that they be deleted.

17-B $ UFI Update the UI database of known applications.

username: ‹username›
password: ‹password›

Enter the following line:

```
INSERT INTO ROLAPP VALUES
  ('‹role in capital letters›', 'SDRPT1ZZZZ›');
EXIT
```

The following steps update the NTM database.

18-B $ EDIT/EDT ACTTBL.DAT Insert new lines as follows:

RPT1ZZZZ1
‹RP-Main process name padded to 8 characters with Z's›1

19-B $ EDIT/EDT APITBL.DAT

Insert new lines as follows:

SDRPT1ZZZZT1V1
GR‹RP-Main process name padded to 8 characters with Z's›T1V1

20-B $ EDIT/EDT APTTBL.DAT

Insert new lines as follows:

RPT1ZZZZ0599010320000010
‹RP-Main name padded to 8 characters with Z's›9999010120001130

21-B $ CREATE RPT1.DAT Create an empty file.
^Z Control Z.

22-B $ VT100

Start up the VT100 device driver.

If the User Interface system has been installed at your site with a different device driver, then this step is amended as appropriate.

23-B Fill in the items on the IISS logon screen as follows:

Username: ‹your username›
Password: ‹your password›
Role : ‹your role›
Press ‹ENTER›

24-B Fill in the items on the FUNCTION screen as follows:

Function: SDDEFINEAP
Press ‹ENTER›

25-B Fill in the fields on the screen that is displayed as follows:

report: SDRPT1ZZZZ
description: REPORT WRITER
host: VAX
Press ‹ENTER›
Press ‹PF4›

26-B Fill in the function field on the screen as follows (Note that the report will be sent to the file rpt1.dat) :

Function : SDRPT1ZZZZ
Device Type : SDPRINTZZZ
Device Name : RPT1.DAT
Press ‹ENTER›

27-B When the function screen is displayed:

Press ‹PF4›

APPENDIX A

TEST REPORT RPT1.FDL

The following is the file RPT1.FDL which is the source file for the Report Writer Unit Test.

```
create report rpt1

                                        /* start up condition */

on (startup)

                                        /* select action */

  SELECT 'qtype(1).stype(1).dbname'   = DB.DB_NAME
     'qtype(1).stype(1).setid'    = SM.SET_ID
     'tstrpt.rtownid' = RS.RT_ID_OF_OWNER
     'tstrpt.rtmemid' = SM.RT_ID_OF_MEMBER
     'tstrpt.reqopt'  = SM.REQ_MEM_IND
 FROM DATA_BASE DB,
      RECORD_TYPE RT,
      RECORD_SET RS,
      SET_TYPE_MEMBER SM
 WHERE  DB.DB_ID = RT.DB_ID AND
    RT.DB_ID = RS.DB_ID AND
    RT.RT_ID = RS.RT_ID_OF_OWNER AND
    RS.DB_ID = SM.DB_ID AND
    RS.SET_ID = SM.SET_ID

                                        /* set action */
  set 'tstrpt.tstnum;' = 123
  set 'tstrpt.tstchar;' = "hello"
                                    /* present form action */
  present tstrpt

                                        /* overflow condition */

on (overflow by 'tstrpt.qtype(1).stype(1);')

                                /* present qualified name action */
  present 'tstrpt.qtype;'

                                        /* change condition */
```

```
on (change 'qtype(1).stype(1).dbname')
                                              /* page action */
  page
  present tstrpt

CREATE FORM tstrpt
    size 80 by 23

    PROMPT AT 1 35
            "REPORT OF CODASYL SET TYPES"
    PROMPT AT 3 2 "DB_NAME"
    PROMPT AT 4 2 "__________"

    PROMPT AT 3 13 "SET_ID"
    PROMPT AT 4 13 "____________________"

    ITEM pdate                              /* calculated field */
    DISPLAY AS text
    AT 1 2
    SIZE 10
    VALUE '._date;'

    ITEM ppage
    DISPLAY AS text
    AT 1 70
    VALUE '._pageno;'
    PROMPT AT left "Page"

    item tstnum
    at 2 2
    size 5
    display as text

    item tstchar
    at right of tstnum
    size 5
    display as text

    form qtype (* h 0)          /* open ended array of forms */
    at 5 1
    size 38 by *

    ITEM rtownid
    DISPLAY AS text
    SIZE 20
```

```
    ITEM rtmemid
    DISPLAY AS text
    SIZE 20

    ITEM reqopt
    DISPLAY AS text
    SIZE 1

create form qtype
    size 38 by 19
    FORM STYPE (* VERTICAL 0)
    AT 1 1
    SIZE 38 BY 1

CREATE FORM stype

    ITEM dbname                          /* nonduplicated values */
    nodup
    DISPLAY AS text
    AT 1 2
    SIZE 10

    ITEM setid
    DISPLAY AS text
    AT 1 13
    SIZE 20
```

APPENDIX B

SAMPLE OUTPUT OF REPORT, RPT1.DAT

```
 7/30/85                REPORT OF CODASYL SET TYPES    Page    1
123   hello
DB_NAME     SET_ID
___________ ____________________                    ____________________
PIOS        UBMF01-PRTMS-PLT                        UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN                        UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN                        UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
            UBMF01-PLTLOC-BN
```

7/30/85 REPORT OF CODASYL SET TYPES Page 2
123 hello

DB_NAME	SET_ID
MCMM	STORES_WO_RESOURCE
	CONTROLLED_BY
	STORED_WO_RESOURCE
	LOADED_WITH
	IDENT_FOR_PROD

7/30/85 REPORT OF CODASYL SET TYPES Page 3
123 hello

DB_NAME	SET_ID
DBD1	ST3
	ST1
	ST2

END

8-87

DTIC